

Настройка и проверка распределенных функций на маршрутизаторах Cisco 75xx и 76xx

Содержание

[Введение](#)

[Предварительные условия](#)

[Требования](#)

[Используемые компоненты](#)

[Условные обозначения](#)

[Распределенные функции](#)

[Распределенный протокол MLPPP](#)

[Распределенный LFI](#)

[Различия между dMLP и dLFioLL](#)

[Распределенный MLFR](#)

[Распределенный DDR](#)

[Распределенные предварительные условия функций и ограничения](#)

[Количество Бандлов и Ссылок и Требований к памяти](#)

[MLPPP программного и аппаратного обеспечения или MLFR на 7600 линейных платах SIP](#)

[Жизнь пакета](#)

[Путь данных gx](#)

[Путь данных tx](#)

[Повторная сборка](#)

[Настройка, проверка и отладка распределенных функций](#)

[Настройка и dMFR Проверки](#)

[Настройка и dMLP/dLFioLL Проверки](#)

[Настройка и dLFioFR Проверки и dLFioATM](#)

[Настройка и dDDR Проверки](#)

[Отладка dMLP и dDDR](#)

[Вопросы и ответы](#)

[Усовершенствования отладки](#)

[Дополнительные сведения](#)

[Введение](#)

Этот документ помогает вам понимать, настраивать, и проверять эти функции:

- Распределенный многоканальный протокол "точка-точка" (dMLP)
- Фрагментация распределенного канала и чередующийся (LFI)

- Распределенный LFI по выделенной линии (dLFloLL)
- Распределенный LFI по Frame Relay (dLFloFR)
- Распределенный LFI по ATM (dLFloATM)
- Различия между Распределенным MLP (dMLP) и dLFloLL
- Распределенный многоканальный Frame Relay (dMLFR)
- Распределенная технология DDR

Предварительные условия

Требования

Читатели данной документации должны ознакомиться с распределенными функциями Cisco 7500/7600/6500.

Используемые компоненты

Сведения, содержащиеся в данном документе, касаются следующих версий программного обеспечения и оборудования:

- Весь Cisco 7500 и 7600 платформ **Примечание:** Сведения в этом документе также применяются к 6500 платформам.
- Соответствующие версии программного обеспечения Cisco IOS, которые приводит эта таблица:

Распределенная поддержка функций каждого ответвления и платформы

Функция	Адаптеры портов (PAs) ¹ поддерживаемое	7500 Платформ		7600 Платформ	
		Версии основного ПО Cisco IOS	(Промежуточные) Cisco IOS Release	Версии основного ПО Cisco IOS	(Промежуточные) Cisco IOS Software Release
dMLP	PA-4T PA chan + PA-8T	12.0T 12.0S 12.1T 12.1S 12.2T 12.2S 12.3T 12.3S 12.2T 12.2S	12.0 (3) T и позже 12.0 (9) S и позже	12.2SX 12.1E ²	—

		12.1E 2			
dLFloLL	PA-4T PA chan + PA-8T	12.2 T 12.3 12.3 T 12.0 S	12.2 (8) T и позже 12.0 (24) S и позже	12.2 SX	12.2 (17) SXB и позже
dLFloFR	PA-4T PA chan + PA-8T	12.2 T 12.3 12.3 T	12.2 (4) T3 и позже	12.2 SX	12.2 (17) SXB и позже
dLFloAT M	PA-A6 PA-A3	12.2 T 12.3 12.3 T	12.2 (4) T3 и позже	12.2 SX	12.2 (17) SXB и позже
dMLFR	PA-4T PA chan + PA-8T	12.0 S 12.3 T	12.0 (24) S и позже 12.3 (4) T и позже	12.2 SX	12.2 (17) SXB и позже
QoS на dMLP	PA-4T PA chan + PA-8T	12.0 S 12.2 T 12.3 12.3 T	12.0 (24) S и позже 12.2 (8) T и позже	12.2 SX	12.2 (17) SXB и позже
MPLS на MPLS dMLP на dLFloLL	PA-4T PA chan + PA-8T	12.2 T 12.3	12.2 (15) T11 и более поздние 12.3 (5a) и позже	12.2 SX	12.2 (17) SXB и позже
Распределенный DDR	PA-MC- xT1 PA- MC-xE1 PA-MC- xTE1 PA- MCX- xTE1	12.3 T	12.3 (7) T и позже	—	—

Примечание: Узнайте об этой информации:

1. Этот PAs поддерживает распределенные функции: CT3IPPA-MC-T3PA-MC-2T3 +PA-MC-E3PA-MC-2E1PA-MC-2T1PA-MC-4T1PA-MC-8T1PA-MC-8E1PA-MC-8TE1 +PA-MC-STM-1
2. Cisco IOS Software Release 12.1E поддерживает эти функции и на 7500 и на 7600

платформах.

Сведения, представленные в этом документе, были получены от устройств, работающих в специальной лабораторной среде. Все устройства, описанные в этом документе, были запущены с чистой (стандартной) конфигурацией. В рабочей сети необходимо изучить потенциальное воздействие всех команд до их использования.

Условные обозначения

[Дополнительные сведения об условных обозначениях см. в документе Технические рекомендации Cisco. Условные обозначения.](#)

Распределенные функции

Эти функции объяснены в этом документе:

- Распределенный MLP
- Распределенный LFI
- Распределенный LFI по выделенной линии
- Распределенный LFI по Frame Relay
- Распределенный LFI по ATM
- Различия между dMLP и dLFIoLL
- Распределенный MLFR
- Распределенный номеронабиратель
- Платформы и Линейные платы, которые поддерживают распределенные функции

Распределенный протокол MLPPP

Распределенный Многоканальный протокол "точка-точка" (dMLP), функция позволяет вам комбинировать полные или дробные линии T1/E1 в линейной карте (VIP, FlexWAN) на Cisco 7500 или маршрутизаторе серии "7600" в связку (bundle), которая имеет объединенные ссылки пропускной способности нескольких интерфейсов T1/E1. Вы используете распределенный Пучок MLP, чтобы сделать это. Пользователь может выбрать количество связок (bundle) в маршрутизаторе и количество ссылок на связку (bundle). Это позволяет пользователю увеличивать пропускную способность соединений сети кроме того одиночной линии T1/E1 без потребности купить линию T3. В не-dMLP все пакеты коммутированы Процессором маршрута (RP); поэтому, эта реализация влияет на производительность RP с высокой загрузкой ЦП только для нескольких линий T1/E1 рабочий MLP. С dMLP увеличено общее число связок (bundle), которые могут быть обработаны на маршрутизаторе, поскольку путь данных обработан и ограничен с методической точностью ЦП карты и память. dMLP позволяет вам связывать дробный T1/E1, запускающийся с DS0 (64 кбит/с) и далее.

Распределенный LFI

Поддержки характеристик dLFI транспорт трафика в реальном времени (такие как голос) и нетрафик реального времени (такие как данные) на Frame Relay малого быстрогодействия и виртуальных каналах ATM (VC) и на выделенных линиях, не вызывая избыточную задержку к трафику в реальном времени.

Эта опция реализована с помощью протокола PPP (MLP) по Frame Relay, ATM и выделенным линиям. Функция фрагментирует большой пакет данных в последовательность меньших фрагментов, чтобы позволить чувствительным к задержке пакетам в реальном времени и пакетам не в реальном времени поделиться той же ссылкой. Фрагменты тогда чередуются с пакетами в реальном времени. На получающей стороне ссылки повторно собраны фрагменты, и пакет восстановлен.

Функция dLFI часто полезна в сетях, которые передают трафик в реальном времени через Distributed Low Latency Queuing (такой как голос), но имеют проблемы связанные с пропускной способностью. Это задерживает трафик в реальном времени из-за транспорта больших, меньшего количества критичных по времени пакетов данных. Можно использовать функцию dLFI в этих сетях, для разборки больших пакетов данных во множественные сегменты. Пакеты трафика в реальном времени тогда могут быть переданы между этими сегментами пакетов данных. В этом сценарии трафик в реальном времени не испытывает длинную задержку, в то время как это ждет низкоприоритетных пакетов данных для пересечения сети. Пакеты данных повторно собраны в получающей стороне ссылки, таким образом, данные отправлены неповрежденные.

Размер фрагмента ссылки вычислен на основе задержки фрагмента на многоканальном соединении, настроенном с **ppp multilink fragment delay n** команда, где:

```
fragment size = bandwidth × fragment-delay / 8
```

Этот размер фрагмента представляет полезную нагрузку IP только. Это не включает байты инкапсуляции (размер фрагмента = вес – байты инкапсуляции). Сроки “вес” и “размер фрагмента” как замечены в выходных данных команды **show ppp multilink** на RP. Если задержка фрагмента не настроена, размер фрагмента по умолчанию вычислен для максимальной задержки фрагмента 30.

Примечание: Со ссылками переменной пропускной способности выбранный размер фрагмента основывается на ссылке с наименьшим количеством пропускной способности.

[Распределенный LFI по выделенной линии](#)

Функция dLFloLL расширяет фрагментацию распределенного канала и чередующуюся функциональность к выделенным линиям. Распределенный LFI настроен с командой **ppp multilink interleave** на интерфейсе многоканальной группы. Желательно, чтобы вы использовали распределенный LFI на многоканальных интерфейсах с пропускной способностью меньше чем 768 кбит/с. Это вызвано тем, что задержка сериализации на 1500 пакеты в 1 байт для пропускной способности, больше, чем 768 кбит/с, в пределах допустимой задержки и не должны быть фрагментированы.

[Распределенный LFI по Frame Relay](#)

Функцией dLFloFR является расширение Протокола PPP по Frame Relay (MLPoFR) функция. MLP используется для фрагментации. Эта функция подобна FRF.12, который поддерживает фрагментацию и может чередовать высокоприоритетные пакеты через Организацию очереди с малой задержкой.

Команда **ppp multilink interleave** на Virtual-template требуется, чтобы позволять чередоваться на связанном интерфейсе виртуального доступа. В дополнение к включению распределенного CEF, включающего последовательный интерфейс, эта команда является предпосылкой для распределенного LFI для работы.

Примечание: Пока вы не используете Frame Relay для Межсетевого взаимодействия ATM, рекомендуется использовать FRF.12, а не dLFIoFR, потому что использование пропускной способности лучше с FRF.12

Распределенный LFI по ATM

Функцией dLFIoATM является расширение Протокола PPP по ATM (MLPoATM) функция. MLP используется для фрагментации.

Команда ppp multilink interleave на Virtual-template требуется, чтобы позволять чередоваться на связанном интерфейсе виртуального доступа. В дополнение к включению распределенного CEF, включающего последовательный интерфейс, эта команда является предпосылкой для распределенного LFI для работы.

С dLFIoATM очень важно, чтобы вы выбрали размер фрагмента, который делает пакеты для помещений в ячейки ATM таким способом, которым они не вызывают ненужное заполнение в ячейках ATM. Например, если выбранный размер фрагмента составляет 124 байта, это означает, что полезная нагрузка IP 124 байтов наконец пошла бы как $124 + 10$ (Заголовок MLP) + 8 (Заголовок SNAP) = 142 байта. Следует отметить, что первый фрагмент вышел бы с $124 + 10 + 2$ (Первый Размер заголовка PID Фрагмента) + 8 = 144 байта. Это означает, что этот пакет будет использовать три ячейки ATM для передачи информационного наполнения и, следовательно, использовал бы ячейку, упакованную наиболее эффективно.

Различия между dMLP и dLFIoLL

dMLP не поддерживает фрагментацию на передающей стороне, тогда как dLFIoLL делает.

Примечание: Чередование и фрагментация, используемая с несколькими ссылками в многоканальном соединении для голосового трафика, не гарантирует, что голосовой трафик, полученный через сложные соединения в связке (bundle), будет получен в заказе. Корректный заказ голоса обрабатывается в верхних уровнях.

Распределенный MLFR

Распределенная функция MLFR представляет функциональность на основе Форума Frame Relay Многоканальный Frame Relay UNI / Соглашение по реализации (FRF.16) NNI поддерживающему линейную карту Cisco 7500 и маршрутизаторам серии "7600".

Распределенная функция MLFR предоставляет экономически эффективный способ увеличить пропускную способность для конкретных приложений, потому что это позволяет многочисленным последовательным каналам быть объединенными в одиночную связку (bundle) пропускной способности. MLFR поддерживается на Сетевых интерфейсах пользователя (UNI) и Интерфейсах сеть-сеть (NNI) в Сетях Frame Relay.

Связка (bundle) составлена из многочисленных последовательных каналов, названных связками каналов. Каждая связка каналов в связке (bundle) соответствует физическому интерфейсу. Связки каналов невидимы для уровня канала передачи данных Frame Relay, таким образом, функциональность Frame Relay не может быть настроена на этих интерфейсах. Функциональность Реле Обычного кадра, что вы хотите примениться к этим ссылкам, должна быть настроена на групповом интерфейсе. Связки каналов видимы к одноранговым устройствам.

Распределенный DDR

Распределенная функция DDR позволяет распределенную коммутацию на интерфейсах номеронабирателя. Без этой функции весь трафик наборного (телефонный) доступа должен плыть на плоскодонке к хосту к коммутации. С ним только управляющие пакеты передаются до RP, тогда как решение о коммутации сделано на самих VIP после того, как было установлено соединение.

И Устаревшая Конфигурация программы набора номера и Конфигурация профиля DDR поддерживаются только с инкапсуляцией PPP. MLP на Интерфейсах номеронабирателя также поддерживается. QoS не поддерживается с распределенной коммутацией на Интерфейсах номеронабирателя.

Распределенные предварительные условия функций и ограничения

Предварительные условия

Это общие предварительные условия для всех этих распределенных функций:

- Distributed Cisco Express Forwarding (dCEF) коммутация должен быть включен глобально.
- коммутация DCEF должна быть включена на задействованном последовательном интерфейсе, которые являются частью Пучка MLP.
- коммутация DCEF должна быть включена на физическом соединении интерфейсов dLFloATM и dLFloFR.
- Конфигурация чередования требуется, чтобы распределять LFloFR и LFloATM.
- Настройте необходимую пропускную способность на Виртуальном интерфейсе для интерфейсов dLFloATM и dLFloFR.
- Когда отладки PPP включены на RP, вы могли бы наблюдать сообщение `MLP: Forwarded to wrong interface` относительно Процессора переключателей маршрута (RSP). Поскольку это сообщение сбивает с толку и нежелательно — особенно, если сообщение для пакетов протокола CDP — необходимо настроить, **no cdp enable** на участвующих соединениях связки (bundle).
- Всем участвующим соединениям связки (bundle) нужно включить поддержку активности.

Ограничения

Это общие ограничения для всех этих распределенных функций:

- Линии T1 and E1 не могут быть смешаны в связке (bundle).
- Максимальная поддерживаемая дифференциальная задержка составляет 30 мс.
- Все линии в связке (bundle) должны находиться на адаптере того же порта (PA).
- Аппаратное сжатие не поддерживается.
- VIP или FlexWAN CEF ограничены IP только; все другие протоколы передаются RSP.
- Фрагментация не поддерживается на передающей стороне для dMLP и dMLFR.
- Многие более старые механизмы организации очередей не поддерживаются dLFlo. Эти механизмы включают: Справедливая очередь на интерфейсе виртуального шаблона Random-detect на интерфейсе виртуального шаблона Настраиваемая

- организация очереди
- Постановка в очередь с установлением приоритета
- Организация очереди с весами, случайное обнаружение (DWRED) и постановка в очередь с установлением приоритета могут быть настроены в политике трафика с Modular QoS CLI.
- Только одна ссылка на Пучок MLP поддерживается при использовании dLFloFR или dLFloATM. Если несколько ссылок используются в Пучке MLP при использовании dLFloFR или dLFloATM, dLFI автоматически отключен. При использовании dLFI по выделенным линиям несколько ссылок могут быть настроены с dLFI в Пучке MLP.
- С dLFloATM только поддерживаются aal5snap и aal5mux. Aal5nlpid инкапсуляции и aal5ciscopp не поддерживаются.
- Только Передача голоса по IP поддерживается; Голос по Frame Relay и Передаче голоса по ATM не поддерживается функцией dLFI.
- Конфигурация Протокола CRTP не должна быть настроена на многоканальном интерфейсе при использовании этой комбинации признаков: Многоканальный интерфейс с включенным LFIM Многоканальное соединение имеет несколько участвующие соединения Политика QoS с функцией приоритета включена на многоканальном интерфейсе

С dMLP и конфигурацией dLFI, приоритеты пакета не несут заголовок MLP и порядковый номер, и MLP распределит приоритеты пакета через все участвующие соединения. В результате пакеты, которые сжаты CRTP, могут поступить не в порядке в принимающий маршрутизатор. Это мешает CRTP распаковывать заголовок пакета и вынуждает CRTP отбросить пакеты.

Рекомендации

Рекомендуется, чтобы участвующие соединения в связке (bundle) имели ту же пропускную способность. Если вы добавите неравные пропускные способности соединения к связке (bundle), то она приведет к партии пакетного переупорядочения, которое заставит полную пропускную способность связки (bundle) уменьшаться.

VIP2-50 (с SRAM на 8 МБ) или выше рекомендуют использоваться с этими распределенными функциями.

Количество Бандлов и Ссылок и Требований к памяти

См. [распределенный многоканальный протокол "точка-точка" для маршрутизаторов Cisco серии 7500](#).

MLPPP программного и аппаратного обеспечения или MLFR на 7600 линейных платах SIP

MLP и MLFR могут быть или программным обеспечением - или аппаратными средствами - основанный. В основанном MLP или MLFR аппаратных средств, Freedm предоставляет функциональные средства с многоканальной структурой, и заголовки MLP добавлены Микросхемой Freedm. В основанном MLP или MLFR программного обеспечения, ЦП линейной карты SIP предоставляет функциональные средства с многоканальной структурой (который подобен VIP и реализациям FlexWAN).

Это ограничения, и условия выполнить аппаратные средства базировали MLP или MLFR.

- Может быть только максимум 336 связок (bundle) на линейную карту и 168 связок (bundle) на Оценку положения безопасности (SPA) (Freedm).
- Может быть только максимум 12 DS1/E1 на связку (bundle).
- Все ссылки должны принадлежать тому же SPA (Freedm).
- Все ссылки в связке (bundle) должны работать на той же скорости.
- Размер фрагмента TX может быть 128, 256, или 512. Настроенный размер фрагмента CLI сопоставлен с самым близким поддерживаемым размером фрагмента.


```
IF (0 < cli_fragment_size - 6 < 256)
  configured_fragment_size = 128
Else IF (cli_fragment_size < 512)
  configured_fragment_size = 256
Else
  configured_fragment_size = 512
```
- Размер фрагмента RX может быть 1 к 9.6 КБ.
- Собственный формат Cisco не может поддерживаться (MLFR).

В аппаратном LFI, если существует только одна ссылка в связке (bundle) и если это - DS1/E1 тогда, Фрагментация и Чередование будут сделаны Freedm.

Выходные данные **show ppp multilink** показывают, работает ли аппаратная реализация.

```
Multilink1, bundle name is M1
Bundle up for 00:14:51
Bundle is Distributed

0 lost fragments, 0 reordered, 0 unassigned
0 discarded, 0 lost received, 1/255 load
Member links: 1 active, 0 inactive (max not set, min not set)
Se6/1/0/1:0, since 00:14:51, no frags rcvd
Distributed fragmentation on. Fragment size 512. Multilink in Hardware.
```

Если многоканальное программное обеспечение - основанное тогда, **выходные данные show ppp multilink** не будут иметь Multilink in Hardware в выходных данных.

Жизнь пакета

Путь данных rx

1. Пакет получен драйвером.
2. Инкапсуляция проверена: следующим образом
 Основная инкапсуляция: В dMLP тип инкапсуляции для входного интерфейса является ET_PPP. В dMLFR тип инкапсуляции для входного интерфейса является ET_FRAME_RELAY. В dLFloLL тип инкапсуляции для входного интерфейса является ET_PPP. В dLFloFR тип инкапсуляции для входного интерфейса является ET_FRAME_RELAY. В dLFloATM тип инкапсуляции для входного интерфейса является ET_ATM. В dDialer тип инкапсуляции является ET_PPP.
 Дополнительная обработка инкапсуляции: Для ET_PPP пронюхивается NLPID. Для dMLP NLPID является MULTILINK. Для dLFloLL существует две вещи рассмотреть: Пакеты VoIP — Они не имеют заголовка MLP и имеют NLPID, который указывает на IP. Пакеты данных — NLPID является MULTILINK. Для dDialer пакеты не будут иметь заголовка MLP и иметь NLPID, которые указывают на IP.
Примечание: В этом случае можно настроить dCRTP (Распределенный Протокол Протокола CRTP). Если так, заголовок распакован перед дальнейшей обработкой.
3. Для ET_FRAME_RELAY, если ссылка, на которой получен пакет, настроена для dMLFR

тогда, пакет обработан для dMLFR

4. Для dLFloFR и dLFloATM, тип инкапсуляции является ET_FRAME_RELAY и ET_ATM, соответственно; но в этом существует заголовок PPP. Заголовок PPP, как с dLFloLL, укажет, является ли пакет голосовым пакетом или пакетом данных. Если dCRTP настроен, заголовок распакован перед дальнейшей обработкой. Голосовые пакеты сразу коммутированы. Пакет фрагментированных данных должен будет быть повторно собран, прежде чем он будет коммутирован. С ET_PPP вы могли бы столкнуться с пакетами Канала "PPP"; и с ET_FRAME_RELAY, вы могли бы столкнуться с управляющими пакетами MLFR. Все эти управляющие пакеты плывутся на плоскодонке к RP для обработки.
5. В зависимости от вышеупомянутого декодирования пакет проверен для типа коммутации, это требует. Тип канала определит, должен ли пакет быть коммутирован IP или коммутирован MPLS. Пакеты тогда даны соответствующим функциям коммутации.
6. Со связыванием в сочетании с распределенными функциями украден турбо вектор быстрой коммутации IP. Это сделано, потому что пакет получен на участвующем соединении; однако, это должно быть, рассматривал таким образом, что это получено на связке (bundle). Также необходимо проверить для управляющих пакетов, которые плывутся на плоскодонке к хосту. В основном в dMLFR, существуют пакеты Интерфейса локального управления (LMI), которые *не* являются управляющими пакетами MLFR. Для них используется другая часть пространства номера dLCI. Каждый раз, когда dLCI декодируется для падения в этом пространстве, пакет плывется на плоскодонке до хоста, потому что это распознано, чтобы быть пакетом LMI. Пакеты VoIP (помещенный в очередь в Очереди с низкой задержкой) просто выключены без добавления заголовка MLP. Когда пакеты фрагментированных данных получены, распределенные функции могут получить и повторно собрать пакеты. Процесс повторной сборки объяснен в последующем разделе. Если пакет должен быть коммутирован меткой, его тогда передают к подпрограмме коммутации на основе тэгов в dMLP. В противном случае, если это должно быть коммутировано IP, это передают к подпрограмме IP-коммутации. **Примечание:** Все пакеты не-IP плывутся на плоскодонке для хостинга в dMLFR.
7. IP: функция IP-коммутации характерна для всех пакетов. Это делает в основном три вещи: Сделайте необходимую обработку пакетов, в случае, если настроены любые функции. Кроме того, когда Распределенный Номерабираетел используется, сделайте обновления счетчика простоя здесь, когда получен "вызывающий интерес пакет". См. [dialer idle-timeout \(interface\)](#), [dialer fast-idle \(interface\)](#) и [Настройку Профиль DDR](#) для подробных данных параметра конфигурации **счетчика простоя**. На 75xx маршрутизаторы, смежность укажет `tx_acc_ptr` для исходящего интерфейса. Если исходящий интерфейс является интерфейсом виртуального доступа, `tx_acc_ptr` является NULL. В этом случае согласуйте инкапсуляцию и получите `tx_acc_ptr` от `fib hwidb`. Этот поиск и инкапсуляция договариваются, необходимо в dLFloFR и dLFloATM. В dLFloLL ссылка рассматривается как часть Многоканального соединения. **Примечание:** TTL для пакета отрегулирован здесь, и проверка для Фрагментации ip осуществлена. `mci_status` установлен в `RXTYPE_DODIP` для всех пакетов.
8. Со сделанным решением о коммутации пакет готов быть поставленным из интерфейса. Интерфейс проверен, чтобы определить, поддерживает ли он локальный коммутатор. Если это делает, это отослано непосредственно через `fastsend`. В противном случае попытка предпринята к коммутатору `route-cache` это. Обратите

внимание на то, что, в случае, если QoS настроено для интерфейса, вектор локального коммутатора украден QoS. HQF будет ставить в очередь пакет и дальнейший процесс пакет, прежде чем это будет наконец передано из интерфейса. Дело обстоит так с dLFI. Для dLFI, фрагментации и чередования установлен. QoS обрабатывает вызов нашей программы фрагментации и чередует фрагментированные пакеты с голосовыми пакетами, которые будут помещены в очередь в очереди с приоритетами (если LLQ будет настроен). Это гарантирует, что Пакеты VoIP не страдают от задержки, требуемой отправить огромные пакеты данных посредством ссылки.

[Путь данных tx](#)

`vip_dtq_consumer` получает пакет и получает номер интерфейса, от которого это получает `idb`. Подпрограмму `fastsend`, которая соответствует `idb`, вызывают:

i) Fastsend

1. В dMFR `fr_info` структура получена из таблицы, которая совпадает с `if_index` к `fr_info`. Управляющие пакеты просто отосланы. Заголовок фрейма даст dLCI, который поможет вам определять, является ли это пакетом LMI или пакетом данных. Поле `dlci` в заголовке фрейма перезаписано с порядковым номером `dmfr`. Отдельные порядковые номера используются для LMI и пакетов данных. **Примечание:** Отдельные порядковые номера используются для отдельных DLCI.
2. В dMLP управляющие пакеты передаются с параметром `Priority` высоко. С пакетами данных, если dCRTP настроен, сжат заголовок. Заголовок MLP VIP, который включает информацию упорядочения, добавлен и передан из участвующих соединений.
3. В dLFI HQF перехватывает пакеты, которые будут передаваться через интерфейс. Если это - голосовой пакет, голосовой пакет размещен в очередь с приоритетами (если LLQ настроен), и передается из интерфейса без инкапсуляции MLP. С пакетами данных это вызывает код фрагментации dLFI, который возвращает фрагменты к коду QoS, которые тогда чередованы с приоритетным трафиком так, чтобы были встречены требования к задержке голосового трафика. Кроме того, если dCRTP настроен, только заголовок для голосового пакета сжат. Заголовки пакета данных оставляют как они
4. В dDialer пакет классифицирован для сброса таймера простоя выходной ссылки, прежде чем будет отослан пакет. Это сделано после того, как выходная ссылка выбрана, если несколько ссылок связаны с тем же номеронабирателем. Никакой заголовок не добавлен к пакетам номеронабирателя. Таким образом упорядочения и повторно собираются пакетов, не поддерживаются на Интерфейсах номеронабирателя.

Примечание: В dMLP, dDialer, dMLFR и dLFI с несколькими ссылками, физическое соединение, на котором передан трафик, зависит от перегрузки ссылки. Если ссылка переполнена, перемещение к следующей ссылке и так далее. (dMLFR, dMLP без QoS и функции dDialer также выбирают ссылки на основе количества байтов, помещаемых на ссылку. Если текущая ссылка уже передала свою долю байтов на не основе кольцевого алгоритма Round robin, это выбирает следующую ссылку. Эта квота решена `frag_bytes` для ссылки. Для задействованных интерфейсов Номеронабирателя `frag_bytes` установлен в значение по умолчанию полосы пропускания интерфейса.)

Примечание: В конфигурациях HQF на интерфейсах выходного VIP HQF крадет `dtq_consumer` вектор. Пакет DMA'd к выходному VIP сначала проходит проверку HQF. Если

QoS настроено на исходящем интерфейсе, HQF умирает для обработки пакета, прежде чем пакет будет fastsent из интерфейса.

Повторная сборка

Простые интерфейсы dDialer не поддерживают повторную сборку и упорядочение. Для включения этого на интерфейсах номеронабирателя MLP по интерфейсам номеронабирателя должен быть настроен. Если это сделано, путь Rx и Tx идентичны путям dMLP. Когда пакеты получены, порядковый номер проверен против ожидаемого порядкового номера.

- Если совпадают порядковые номера: Если пакет является нефрагментированным пакетом тогда, повторная сборка не требуется. Продолжите дальнейшие шаги коммутации. Если пакет является фрагментом, то проверьте начинание и конечные биты и создайте пакет как и когда получены фрагменты.
- Если не совпадают порядковые номера: Если порядковый номер в ожидаемом окне порядковых номеров, тогда помещает его в сортированный "неназначенный список фрагмента". Позже, когда ожидаемый порядковый номер не получен, этот список проверен, в случае, если пакет был сохранен здесь. Если порядковый номер не в окне, сбросьте от него, и отчёт "получил потерянный фрагмент". Если таймаут происходит позже при ожидании этого пакета, получатель повторно синхронизируется, и это начинается снова с полученного следующего пакета.

Во всех тех случаях правильно упорядоченный поток пакетов передается из этого интерфейса. Если фрагменты получены, полный пакет сформирован и затем отослан.

Настройка, проверка и отладка распределенных функций

Этот раздел покрывает **показ** и **команды отладки**, которые доступны, чтобы проверить и отладить каждую из распределенных функций.

Настройка и dMFR Проверки

Пример конфигурации MFR

```
Multilink1, bundle name is M1
Bundle up for 00:14:51
Bundle is Distributed

0 lost fragments, 0 reordered, 0 unassigned
0 discarded, 0 lost received, 1/255 load
Member links: 1 active, 0 inactive (max not set, min not set)
Se6/1/0/1:0, since 00:14:51, no frags rcvd
Distributed fragmentation on. Fragment size 512. Multilink in Hardware.
```

Примечание: Интерфейс MFR походит на другой Интерфейс FR и следовательно поддерживает большую часть конфигурации FR.

```
Multilink1, bundle name is M1
Bundle up for 00:14:51
Bundle is Distributed

0 lost fragments, 0 reordered, 0 unassigned
0 discarded, 0 lost received, 1/255 load
```

Member links: 1 active, 0 inactive (max not set, min not set)
Se6/1/0/1:0, since 00:14:51, no frags rcvd
Distributed fragmentation on. Fragment size 512. Multilink in Hardware.

[Проверьте состояние связки MFR на RP](#)

```
show frame-relay multilink
```

```
Bundle: MFR1, State = up, class = A, fragmentation disabled  
BID = MFR1  
Bundle links:  
Serial5/0/0/3:0, HW state = up, link state = Add_sent, LID = Serial5/0/0/3:0  
Serial5/0/0/2:0, HW state = up, link state = Up, LID = Serial5/0/0/2:0  
Serial5/0/0/1:0, HW state = up, link state = Up, LID = Serial5/0/0/1:0
```

Это указывает, что два интерфейса добавлены правильно, и один интерфейс еще не выполнил согласование о Сообщениях MLFR lip.

Для получения дополнительных сведений о связке (bundle) MFR и Участвующих соединениях выполните эту команду:

```
show frame-relay multilink mfr1 detailed
```

```
Bundle: MFR1, State = up, class = A, fragmentation disabled  
BID = MFR1  
No. of bundle links = 3, Peer's bundle-id = MFR1  
Rx buffer size = 36144, Lost frag timeout = 1000  
Bundle links:  
Serial5/0/0/3:0, HW state = up, link state = Add_sent, LID = Serial5/0/0/3:0  
Cause code = none, Ack timer = 4, Hello timer = 10,  
Max retry count = 2, Current count = 0,  
Peer LID = , RTT = 0 ms  
Statistics:  
Add_link sent = 35, Add_link rcv'd = 0,  
Add_link ack sent = 0, Add_link ack rcv'd = 0,  
Add_link rej sent = 0, Add_link rej rcv'd = 0,  
Remove_link sent = 0, Remove_link rcv'd = 0,  
Remove_link_ack sent = 0, Remove_link_ack rcv'd = 0,  
Hello sent = 0, Hello rcv'd = 0,  
Hello_ack sent = 0, Hello_ack rcv'd = 0,  
outgoing pak dropped = 0, incoming pak dropped = 0  
Serial5/0/0/2:0, HW state = up, link state = Up, LID = Serial5/0/0/2:0  
Cause code = none, Ack timer = 4, Hello timer = 10,  
Max retry count = 2, Current count = 0,  
Peer LID = Serial6/1/0/2:0, RTT = 32 ms  
Statistics:  
Add_link sent = 0, Add_link rcv'd = 0,  
Add_link ack sent = 0, Add_link ack rcv'd = 0,  
Add_link rej sent = 0, Add_link rej rcv'd = 0,  
Remove_link sent = 0, Remove_link rcv'd = 0,  
Remove_link_ack sent = 0, Remove_link_ack rcv'd = 0,  
Hello sent = 7851, Hello rcv'd = 7856,  
Hello_ack sent = 7856, Hello_ack rcv'd = 7851,  
outgoing pak dropped = 0, incoming pak dropped = 0  
Serial5/0/0/1:0, HW state = up, link state = Up, LID = Serial5/0/0/1:0  
Cause code = none, Ack timer = 4, Hello timer = 10,  
Max retry count = 2, Current count = 0,  
Peer LID = Serial6/1/0/1:0, RTT = 32 ms  
Statistics:  
Add_link sent = 0, Add_link rcv'd = 0,  
Add_link ack sent = 0, Add_link ack rcv'd = 0,
```

```
Add_link rej sent = 0, Add_link rej rcv'd = 0,  
Remove_link sent = 0, Remove_link rcv'd = 0,  
Remove_link_ack sent = 0, Remove_link_ack rcv'd = 0,  
Hello sent = 7851, Hello rcv'd = 7856,  
Hello_ack sent = 7856, Hello_ack rcv'd = 7851,  
outgoing pak dropped = 0, incoming pak dropped = 0
```

[Команды отладки MFR](#)

Эти отладки полезны для решения проблем, где ссылка не становится добавленной к связке (bundle).

```
debug frame-relay multilink control
```

Примечание: Когда определенный интерфейс MFR или Последовательный интерфейс не заданы, это включает отладку для всех ссылок MFR. Если маршрутизатор имеет большое число ссылок MFR, это может быть подавляющим.

Для отладки пакетов MFR, которые получены в RP, а также отладить управляющих деятельность MFR эта отладка полезна:

```
debug frame-relay multilink
```

Примечание: Под большим объемом трафика это может сокрушить ЦП.

[Проверьте состояние пучка dMLFR на LC](#)

```
show frame-relay multilink
```

Примечание: В настоящее время это не доступно на LC, но он будет скоро добавлен. До тех пор используйте `show ppp multilink`.

```
debug frame-relay multilink
```

[Настройка и dMLP/dLFloLL Проверки](#)

[Конфигурация протокола PPP](#)

```
debug frame-relay multilink
```

Пример конфигурации под Последовательным интерфейсом:

```
debug frame-relay multilink
```

Примечание: Команда `ppp chap hostname M1` действительно не означает, что включена Аутентификация CHAP. Если там будет несколькими многоканальными соединениями между теми же двумя маршрутизаторами, строка **M1** в этой команде действует как различитель конечной точки и только требуется. В таком случае все ссылки, которые принадлежат связке (bundle), должны иметь тот же дискриминатор конечной точки, и никакие две ссылки, которые принадлежат другой связке (bundle), не должны иметь тот же дискриминатор конечной точки.

[Параметры произвольной конфигурации](#)

[никакой] ppp multilink interleave

Это позволяет чередоваться на многоканальном соединении. Это работает в сочетании с Modular QoS CLI. В то время как другие пакеты будут фрагментированы и переданы с последовательностью MLP и заголовком, высокоприоритетные пакеты будут переданы без добавления последовательности MLP и заголовка.

Примечание: Когда чередование включено с несколькими ссылками, возможно, что будет переупорядочен трафик с высоким приоритетом. Когда чередование включено или отключено, сброс связки (bundle) требуется, чтобы активировать его в линейной карте.

```
ppp multilink mrru local value
```

Это задает Maximum Receive Unit на многоканальном; пакеты до этого размера будут приняты многоканальным интерфейсом. Размер здесь исключает заголовок MLP.

```
ppp multilink mrru remote value
```

Это задает минимальный MRRU, который должен поддержать удаленный конец. Если MRRU удаленного конца будет меньше, чем это значение, то согласование связки (bundle) откажет.

```
ppp multilink fragment delay seconds
```

Это задает позволенную задержку миллисекунд (мс), вызванный фрагментом данных. Другими словами, значение задержки используется для вычисления максимального позволенного размера фрагмента. Распределенная реализация отличается от реализации Cisco IOS этими способами:

1. Фрагментация не выполнена, пока не включено чередование.
2. Со ссылками переменной пропускной способности выбранный размер фрагмента основывается на наименьшем количестве интерфейса пропускной способности.

```
ppp multilink fragment disable
```

Эта команда не добавляет функциональности в распределенной реализации. Когда чередование включено, фрагментация только происходит; и, когда чередование включено, команда **ppp multilink fragment disable** проигнорирована.

[Проверьте состояние связки dMLP на RP](#)

```
show ppp multilink
```

```
Multilink1, bundle name is M1
Endpoint discriminator is M1
Bundle up for 00:09:09, 1/255 load
Receive buffer limit 24000 bytes, frag timeout 1000 ms
```

Bundle is Distributed

```
0/0 fragments/bytes in reassembly list
0 lost fragments, 0 reordered
0/0 discarded fragments/bytes, 0 lost received
0x9 received sequence, 0x0 sent sequence
```

dLFI statistics:

dLFI Packets	Pkts In	Chars In	Pkts Out	Chars Out
Fragmented	0	0	0	0

```

UnFragmented          9          3150          0          0
Reassembled           9          3150          0          0
Reassembly Drops      0
Fragmentation Drops   0
Out of Seq Frags      0

```

Member links: 2 active, 0 inactive (max not set, min not set)

Se5/0/0/4:0, since 00:09:09, 768 weight, 760 frag size

Se5/0/0/5:0, since 00:09:09, 768 weight, 760 frag size

1. Когда связка (bundle) находится в распределенном режиме, это отображено в выходных данных `show ppp multilink`: Bundle is Distributed. В противном случае тогда связка (bundle) по некоторым причинам не распределена.
2. Когда `ppp multilink interleave` настроен и включен в линейной карте, выходные данные `show ppp multilink` включают `dLFI statistics` где: `Fragmented` — Указывает на количество фрагментов, которые были переданы и получены. `Unfragmented` — Указывает на количество пакетов, которые были переданы или получены без того, чтобы быть фрагментированным. `Reassembled` — Указывает на количество полных пакетов, которые были повторно собраны. Когда чередование не включено, выходные данные похожи на

```

ЭТО:Multilink1, bundle name is M1
Endpoint discriminator is M1
Bundle up for 00:00:00, 0/255 load
Receive buffer limit 24000 bytes, frag timeout 1000 ms
Bundle is Distributed
  0/0 fragments/bytes in reassembly list
  0 lost fragments, 0 reordered
  0/0 discarded fragments/bytes, 0 lost received
  0x0 received sequence, 0x2 sent sequence
Member links: 2 active, 0 inactive (max not set, min not set)
  Se5/0/0/5:0, since 00:00:00, 768 weight, 760 frag size
  Se5/0/0/4:0, since 00:00:03, 768 weight, 760 frag size

```

Размер фрагмента в предыдущем примере составляет 760 байтов.

[Проверьте состояние связки dMLP на LC](#)

```
show ppp multilink
```

```

dmlp_ipc_config_count 24
dmlp_bundle_count 2
dmlp_ipc_fault_count 1
dmlp_il_inst 0x60EE4340, item count 0
0, store 0, hwidb 0x615960E0, bundle 0x622AA060, 0x60579290, 0x6057A29C
1, store 0, hwidb 0x615985C0, bundle 0x622AA060, 0x60579290, 0x6057A29C
2, store 0, hwidb 0x0, bundle 0x0,
3, store 0, hwidb 0x0, bundle 0x0,
4, store 0, hwidb 0x0, bundle 0x0,
5, store 0, hwidb 0x0, bundle 0x0,
6, store 0, hwidb 0x0, bundle 0x0,
7, store 0, hwidb 0x0, bundle 0x0,
8, store 0, hwidb 0x0, bundle 0x0,
9, store 0, hwidb 0x0, bundle 0x0,

```

```
Bundle Multilink1, 2 members
```

```

bundle 0x622AA060, frag_mode 0
tag vectors 0x604E8004 0x604C3628
Bundle hwidb vector 0x6057B198
idb Multilink1, vc 4, RSP vc 4
QoS disabled, fastsend (qos_fastsend), visible_bandwidth 3072
board_encap 0x60577554, hw_if_index 0, pak_to_host 0x0
max_particles 400, mrru 1524, seq_window_size 0x8000

```



```

working_pak 0x0, working_pak_cache 0x0
una_frag_list 0x0, una_frag_end 0x0, null_link 0
rcved_end_bit 1, is_lost_frag 1, resync_count 0
timeout 0, timer_start 0, timer_running 0, timer_count 1
next_xmit_link Serial0/0:3, member 0x3, congestion 0x3
dmlp_orig_pak_to_host 0x603E7030
dmlp_orig_fastsend 0x6035DBC0
bundle_idb->lc_ip_turbo_fs 0x604A7750
  0 lost fragments, 0 reordered, 0 unassigned
  0 discarded, 0 lost received
  0xC3 received sequence, 0x0 sent sequence
  Member Link: 2 active
    Serial0/0:4, id 0x1, fastsend 0x60579290, lc_turbo 0x6057A29C, PTH 0x60579A18, OOF 0
    Serial0/0:3, id 0x2, fastsend 0x60579290, lc_turbo 0x6057A29C, PTH 0x60579A18, OOF 0

```

С dMFR порядковые номера поддержаны на основе HA DLCI с порядковым номером в связке (bundle), используемой для dLCI LMI.

Поле	Описание
dmlp_ipc_config_count	Количество сообщений IPC, полученных LC для многоканального или конфигурации MLFR
dmlp_bundle_count	Количество MLP и MLFR связывает в LC
dmlp_ipc_fault_count	Количество сообщений настройки, которые привели к сбою в LC. Должен быть 0; если это является ненулевым тогда могла бы быть проблема.
tag_vectors	Указывает на idb к tag_optimum_fs и idb к ip2tag_optimum_fs векторам, используемым в коммутации на основе тэгов.
board_encap	Указывает на board_encap вектор, который используется для добавления 2 байтов инкапсуляции платы, если там канализируются ссылки в 7500 платформах. Если ссылка содержит безканальные интерфейсы, должен быть NULL.
max_particles	Максимальное число частиц, которые могут быть проведены в буфере сборки
mrru	Максимальный размер пакета, которые приняты, не рассматривая инкапсуляцию MLP. Не применимый для интерфейса MLFR.
seq_window_size	Максимальный размер окна

	для порядковых номеров
working_pak	Указывает на текущий пакет под повторной сборкой. NULL, если ни один.
working_pak_cache	Указатель на статический пакет, который используется для повторной сборки. Когда первый неполный пакет получен связкой (bundle), это выделено.
una_frag_list	Первая запись в очереди повторной сборки. Если запись не является NULL и не изменяется, это указывает, что таймер не выполняет проблему программного обеспечения.
una_frag_end	Последняя запись в очереди повторной сборки
rcved_end_bit	Указывает, что связка (bundle) получила конечный бит, таким образом, она ищет для начать бита.
is_lost_frag	Если фрагмент объявлен потерянный, истинно. Когда фрагмент с ожидаемой последовательностью получен, это очищено.
resync_count	Указывает на число раз, что получатель был вне синхронизации с передатчиком и должен был повторно синхронизировать начиная с последнего полученного упорядоченного фрагмента.
timeout	Указывает, что тайм-аут пересборки произошел, и пакеты обрабатываются от очереди повторной сборки.
timer_start	Таймер повторной сборки числа раз был запущен
timer_running	Указывает, работает ли таймер повторной сборки.
timer_count	Указывает на число раз, что истек таймер повторной сборки.
next_xmit_link	Ссылка, на которой будет передан следующий пакет
Member	Битовое поле, которое

	указывает на участников, представляет.
Congestion	Поле, не используемое во всех ответвлениях. Указывает, какие участвующие соединения не переполнены.
dmlp_orig_pak_to_host	Вектор, используемый к пакетам избыточного направления к RP.
dmlp_orig_fastsend	Исходный драйвер fastsend перед MLP или MLFR модифицировал fastsend драйвера.
lost fragments	Количество фрагментов, которые были потеряны (получатель не получил эти фрагменты). Когда обновление передается хосту, это периодически очищается.
Reordered	Количество фрагментов, которые были получены из ожидаемого заказа. Когда обновление передается хосту, это периодически очищается.
Discarded	Количество фрагментов сбросило, потому что не мог быть сделан полный пакет
lost received	Количество фрагментов получило, которые, как думали, были потеряны. Это указывает, что задержка связующего звена больше, чем таймаут повторной сборки dMLP 30 мс.

[Настройка и dLFloFR Проверки и dLFloATM](#)

```
class-map voip
  match ip precedence 3
```

```
policy-map llq
  class voip
    priority
```

```
int virtual-templatel
  service-policy output llq
  bandwidth 78
  ppp multilink
  ppp multilink interleave
  ppp multilink fragment-delay 8
```

```
int serial5/0/0/6:0
```

```
encapsulation frame-relay
frame-relay interface-dlci 16 ppp virtual-templatel
!--- Or

int ATM4/0/0
  no ip address
int ATM4/0/0.1 point-to-point
  pvc 5/100
  protocol ppp virtual-template 1
```

[Проверьте Статус Бандла dLFIoFR/ATM на RP](#)

```
show ppp multilink
```

```
Virtual-Access3, bundle name is dLFI
Endpoint discriminator is dLFI
Bundle up for 00:01:11, 1/255 load
Receive buffer limit 12192 bytes, frag timeout 1524 ms
Bundle is Distributed
  0/0 fragments/bytes in reassembly list
  0 lost fragments, 0 reordered
  0/0 discarded fragments/bytes, 0 lost received
  0x0 received sequence, 0x0 sent sequence
dLFI statistics:
      DLFI Packets   Pkts In   Chars In   Pkts Out   Chars Out
      Fragmented           0           0           0           0
      UnFragmented         0           0           0           0
      Reassembled          0           0           0           0
      Reassembly Drops      0
      Fragmentation Drops   0
      Out of Seq Frags      0
Member links: 1 (max not set, min not set)
  vi2, since 00:01:11, 240 weight, 230 frag size
```

Примечание: Связка (bundle) станет распределенной только, когда `ppp multilink interleave` будет настроен под виртуальным шаблоном; без этой команды не будет распределена связка (bundle).

[Проверьте Статус Бандла dLFIoFR/ATM на LC](#)

Проверить dLFI действительно работает правильно над LC, выполните эту команду:

```
show hqf interface
```

```
Interface Number 6 (type 22) Serial0/0:5

  blt (0x62D622E8, index 0, hwidb->fast_if_number=35) layer PHYSICAL
  scheduling policy: FIFO
  classification policy: NONE
  drop policy: TAIL
  blt flags: 0x0

  qsize 0 txcount 3 drops 0 qdrops 0 nobuffers 0
  aggregate limit 16 individual limit 4 availbuffers 16
  weight 1 perc 0.00 ready 1 shape_ready 1 wfq_clitype 0
  visible_bw 64 allocated_bw 64 qlimit_tuned 0 vc_encap 2
  quantum 1500 credit 0 backpressure_policy 0 nothingoncalQ 1

  next layer HQFLAYER_FRAMEDLCI_IFC (max entries 1024)
```

blt (0x62D620E8, index 0, hwidb->fast_if_number=35) layer FRAMEDLCI_IFC
scheduling policy: FIFO
classification policy: NONE
drop policy: TAIL
blt flags: 0x0

qsize 0 txcount 1 drops 0 qdrops 0 nobuffers 0
aggregate limit 16 individual limit 4 availbuffers 16
weight 1 perc 0.00 ready 1 shape_ready 1 wfq_clitype 0
visible_bw 64 allocated_bw 64 qlimit_tuned 0 vc_encap 2
quantum 1500 credit 0 backpressure_policy 0 nothingoncalQ 1

blt (0x62D621E8, index 16, hwidb->fast_if_number=35) layer FRAMEDLCI_IFC
scheduling policy: WFQ
classification policy: PRIORITY_BASED
drop policy: TAIL
frag policy: root
blt flags: 0x0

qsize 0 txcount 2 drops 0 qdrops 0 nobuffers 0
aggregate limit 16 individual limit 4 availbuffers 16
weight 1 perc 0.00 ready 1 shape_ready 1 wfq_clitype 0
visible_bw 64 allocated_bw 64 qlimit_tuned 0 vc_encap 2
quantum 240 credit 0 backpressure_policy 0 nothingoncalQ 1

next layer HQFLAYER_PRIORITY (max entries 256)

blt (0x62D61FE8, index 0, hwidb->fast_if_number=35) **layer PRIORITY**
scheduling policy: FIFO
classification policy: NONE
drop policy: TAIL
frag policy: leaf
blt flags: 0x0

qsize 0 txcount 0 drops 0 qdrops 0 nobuffers 0
aggregate limit 8 individual limit 2 availbuffers 8
weight 0 perc 0.99 ready 1 shape_ready 1 wfq_clitype 0
visible_bw 32 allocated_bw 32 qlimit_tuned 0 vc_encap 2
quantum 240 credit 0 backpressure_policy 0 nothingoncalQ 1

blt (0x62D61CE8, index 1, hwidb->fast_if_number=35) **layer PRIORITY**
scheduling policy: FIFO
classification policy: NONE
drop policy: TAIL
blt flags: 0x0

Priority Conditioning enabled

qsize 0 txcount 0 drops 0 qdrops 0 nobuffers 0
aggregate limit 0 individual limit 0 availbuffers 0
weight 1 perc 0.00 ready 1 shape_ready 1 wfq_clitype 0
visible_bw 0 allocated_bw 0 qlimit_tuned 0 vc_encap 2
quantum 240 credit 0 backpressure_policy 0 nothingoncalQ 1

PRIORITY: bandwidth 32 (50%)
last 0 tokens 1500 token_limit 1500

blt (0x62D61EE8, index 255, hwidb->fast_if_number=35) **layer PRIORITY**
scheduling policy: WFQ
classification policy: CLASS_BASED
drop policy: TAIL
frag policy: MLPPP (1)
frag size: 240, vc encap: 0, handle: 0x612E1320
blt flags: 0x0

qsize 0 txcount 2 drops 0 qdrops 0 nobuffers 0

```
aggregate limit 8 individual limit 2 availbuffers 8
weight 1 perc 0.01 ready 1 shape_ready 1 wfq_clitype 0
visible_bw 32 allocated_bw 32 qlimit_tuned 0 vc_encap 2
quantum 1 credit 0 backpressure_policy 0 nothingoncalQ 1
```

```
next layer HQFLAYER_CLASS_HIER0 (max entries 256)
```

```
blt (0x62D61DE8, index 0, hwidb->fast_if_number=35) layer CLASS_HIER0
scheduling policy: FIFO
classification policy: NONE
drop policy: TAIL
frag policy: leaf
blt flags: 0x0
```

```
qsize 0 txcount 2 drops 0 qdrops 0 nobuffers 0
aggregate limit 8 individual limit 2 availbuffers 8
weight 1 perc 50.00 ready 1 shape_ready 1 wfq_clitype 0
visible_bw 32 allocated_bw 32 qlimit_tuned 0 vc_encap 2
quantum 240 credit 0 backpressure_policy 0 nothingoncalQ 1
```

Должен быть приоритетный уровень и уровень WFQ. Фрагментация будет сделана в конечном уровне WFQ blt.

[Настройка и dDDR Проверки](#)

Распределенный DDR активирован при включении **ip cef**, **распределенного** на глобальной конфигурации и **ip route-cache**, **распределенном** на интерфейсах номеронабирателя.

```
show hqf interface
```

```
Interface Number 6 (type 22) Serial0/0:5
```

```
blt (0x62D622E8, index 0, hwidb->fast_if_number=35) layer PHYSICAL
scheduling policy: FIFO
classification policy: NONE
drop policy: TAIL
blt flags: 0x0
```

```
qsize 0 txcount 3 drops 0 qdrops 0 nobuffers 0
aggregate limit 16 individual limit 4 availbuffers 16
weight 1 perc 0.00 ready 1 shape_ready 1 wfq_clitype 0
visible_bw 64 allocated_bw 64 qlimit_tuned 0 vc_encap 2
quantum 1500 credit 0 backpressure_policy 0 nothingoncalQ 1
```

```
next layer HQFLAYER_FRAMEDLCI_IFC (max entries 1024)
```

```
blt (0x62D620E8, index 0, hwidb->fast_if_number=35) layer FRAMEDLCI_IFC
scheduling policy: FIFO
classification policy: NONE
drop policy: TAIL
blt flags: 0x0
```

```
qsize 0 txcount 1 drops 0 qdrops 0 nobuffers 0
aggregate limit 16 individual limit 4 availbuffers 16
weight 1 perc 0.00 ready 1 shape_ready 1 wfq_clitype 0
visible_bw 64 allocated_bw 64 qlimit_tuned 0 vc_encap 2
quantum 1500 credit 0 backpressure_policy 0 nothingoncalQ 1
```

```
blt (0x62D621E8, index 16, hwidb->fast_if_number=35) layer FRAMEDLCI_IFC
scheduling policy: WFQ
classification policy: PRIORITY_BASED
```

drop policy: TAIL

frag policy: root

blt flags: 0x0

qsize 0 txcount 2 drops 0 qdrops 0 nobuffers 0
aggregate limit 16 individual limit 4 availbuffers 16
weight 1 perc 0.00 ready 1 shape_ready 1 wfq_clitype 0
visible_bw 64 allocated_bw 64 qlimit_tuned 0 vc_encap 2
quantum 240 credit 0 backpressure_policy 0 nothingoncalQ 1

next layer HQFLAYER_PRIORITY (max entries 256)

blt (0x62D61FE8, index 0, hwidb->fast_if_number=35) **layer PRIORITY**

scheduling policy: FIFO

classification policy: NONE

drop policy: TAIL

frag policy: leaf

blt flags: 0x0

qsize 0 txcount 0 drops 0 qdrops 0 nobuffers 0
aggregate limit 8 individual limit 2 availbuffers 8
weight 0 perc 0.99 ready 1 shape_ready 1 wfq_clitype 0
visible_bw 32 allocated_bw 32 qlimit_tuned 0 vc_encap 2
quantum 240 credit 0 backpressure_policy 0 nothingoncalQ 1

blt (0x62D61CE8, index 1, hwidb->fast_if_number=35) **layer PRIORITY**

scheduling policy: FIFO

classification policy: NONE

drop policy: TAIL

blt flags: 0x0

Priority Conditioning enabled

qsize 0 txcount 0 drops 0 qdrops 0 nobuffers 0
aggregate limit 0 individual limit 0 availbuffers 0
weight 1 perc 0.00 ready 1 shape_ready 1 wfq_clitype 0
visible_bw 0 allocated_bw 0 qlimit_tuned 0 vc_encap 2
quantum 240 credit 0 backpressure_policy 0 nothingoncalQ 1

PRIORITY: bandwidth 32 (50%)

last 0 tokens 1500 token_limit 1500

blt (0x62D61EE8, index 255, hwidb->fast_if_number=35) **layer PRIORITY**

scheduling policy: WFQ

classification policy: CLASS_BASED

drop policy: TAIL

frag policy: MLPPP (1)

frag size: 240, vc encap: 0, handle: 0x612E1320

blt flags: 0x0

qsize 0 txcount 2 drops 0 qdrops 0 nobuffers 0
aggregate limit 8 individual limit 2 availbuffers 8
weight 1 perc 0.01 ready 1 shape_ready 1 wfq_clitype 0
visible_bw 32 allocated_bw 32 qlimit_tuned 0 vc_encap 2
quantum 1 credit 0 backpressure_policy 0 nothingoncalQ 1

next layer HQFLAYER_CLASS_HIER0 (max entries 256)

blt (0x62D61DE8, index 0, hwidb->fast_if_number=35) **layer CLASS_HIER0**

scheduling policy: FIFO

classification policy: NONE

drop policy: TAIL

frag policy: leaf

blt flags: 0x0

qsize 0 txcount 2 drops 0 qdrops 0 nobuffers 0

```
aggregate limit 8 individual limit 2 availbuffers 8
weight 1 perc 50.00 ready 1 shape_ready 1 wfq_clitype 0
visible_bw 32 allocated_bw 32 qlimit_tuned 0 vc_encap 2
quantum 240 credit 0 backpressure_policy 0 nothingoncalQ 1
```

Нет никаких других специальных конфигураций для распределенного DDR. Дальнейшая конфигурация придерживается обычной конфигурации DDR.

[Проверьте распределенную маршрутизацию установления соединения по запросу](#)

```
BOX2002# show isdn status
```

```
Global ISDN Switchtype = primary-net5
ISDN Serial3/1/0:23 interface
--- Network side configuration. dsl 0, interface ISDN Switchtype = primary-net5 Layer 1 Status:
ACTIVE Layer 2 Status: TEI = 0, Ces = 1, SAPI = 0, State = MULTIPLE_FRAME_ESTABLISHED
```

```
The ISDN status should be MULTIPLE_FRAME_ESTABLISHED. This means that the physical layer is
ready for ISDN connectivity. Layer 3 Status: 0 Active Layer 3 Call(s) Active dsl 0 CCBs = 0 The
Free Channel Mask: 0x807FFFFFF Number of L2 Discards = 0, L2 Session ID = 6 EDGE# show dialer
```

```
Serial6/0:0 - dialer type = ISDN
Idle timer (120 secs), Fast idle timer (20 secs)
Wait for carrier (30 secs), Re-enable (15 secs)
Dialer state is data link layer up
Time until disconnect 119 secs
Current call connected never
Connected to 54321
```

```
Serial6/0:1 - dialer type = ISDN
Idle timer (120 secs), Fast idle timer (20 secs)
Wait for carrier (30 secs), Re-enable (15 secs)
Dialer state is idle
```

dialer type говорит нам тип используемого номеронабирателя. ISDN подразумевает устаревшую конфигурацию программы набора номера, и PROFILE подразумевает конфигурацию профиля DDR. dialer state указывает на текущее состояние номеронабирателя. Состоянием несвязанного интерфейса номеронабирателя будет idle. Idle timer перезагружен каждый раз, когда замечен представляющий интерес трафик. Если этот таймер когда-нибудь истечет, то интерфейс сразу разъединит. Idle timer является параметром с изменяемой конфигурацией. Для получения дополнительной информации обратитесь к [Настройке Одноранговый DDR с Профилями DDR](#).

```
show ppp multilink
```

```
!--- From LC for dialer profile. dmlp_ipc_config_count 2 dmlp_bundle_count 1 dmlp_il_inst
0x60EE4340, item count 0 0, store 0, hwidb 0x0, bundle 0x0, 1, store 0, hwidb 0x0, bundle 0x0,
2, store 0, hwidb 0x0, bundle 0x0, 3, store 0, hwidb 0x0, bundle 0x0, 4, store 0, hwidb 0x0,
bundle 0x0, 5, store 0, hwidb 0x0, bundle 0x0, 6, store 0, hwidb 0x0, bundle 0x0, 7, store 0,
hwidb 0x0, bundle 0x0, 8, store 0, hwidb 0x0, bundle 0x0, 9, store 0, hwidb 0x0, bundle 0x0,
Bundle Dialer1, 1 member bundle 0x62677220, frag_mode 0 tag vectors 0x604E8004 0x604C3628 Bundle
hwidb vector 0x0 idb Dialer1, vc 22, RSP vc 22 QoS disabled, fastsend (mlp_fastsend),
visible_bandwidth 56 board_encap 0x60577554, hw_if_index 0, pak_to_host 0x0 max_particles 200,
mrru 1524, seq_window_size 0x8000 working_pak 0x0, working_pak_cache 0x0 una_frag_list 0x0,
una_frag_end 0x0, null_link 0 rcved_end_bit 1, is_lost_frag 0, resync_count 0 timeout 0,
timer_start 0, timer_running 0, timer_count 0 next_xmit_link Serial1/0:22, member 0x1,
congestion 0x1 dmlp_orig_pak_to_host 0x603E7030 dmlp_orig_fastsend 0x60381298 bundle_idb-
>lc_ip_turbo_fs 0x604A7750 0 lost fragments, 0 reordered, 0 unassigned 0 discarded, 0 lost
received 0x0 received sequence, 0x0 sent sequence Member Link: 1 active Serial1/0:22, id 0x1,
fastsend 0x60579290, lc_turbo 0x6057A29C, PTH 0x60579A18, OOF 0
```

Показанные переменные совпадают с теми для dMLP.

Отладка dMLP и dDDR

Отладки, доступные в RP

dDDR

```
debug dialer [events | packets | forwarding | map]
```

Выполните эту команду к функциям path управления отладкой как настройка вызова и т.д. Для получения дополнительной информации обратитесь к [debug dialer events](#).

```
debug ip cef dialer
```

Выполните эту команду для отладки связанных с CEF событий номеронабирателя. Для получения дополнительной информации обратитесь к [Dialer CEF](#).

Отладки, доступные в LC

dMLP

Отладка Контрольного пути: **многоканальное событие отладки**

Отладка пути данных: **многоканальные фрагменты отладки**

Путь данных и ошибочная отладка контрольного пути: **многоканальная ошибка отладки**

Отладка dMLP на Линейных платах SIP

Формирование дампа пакетов на основе CI: Пакеты данных и Управляющие пакеты могут быть разгружены на линейных платах на основе контроля ci и последовательности ci.

тестовый hw-module subslot_num формирует дамп ci NUM CI [rx|tx] num_packets_to_dump

CI могут быть получены этим способом:

```
!--- Issue show controller serial interface for CTE1.
```

```
SIP-200-6# show controller serial 6/0/0:0
```

```
SPA 6/0 base address 0xB8000000 efc 1
```

```
Interface Serial6/0/0:0 is administratively down
Type 0xD Map 0x7FFFFFFF, Subrate 0xFF, mapped 0x1, maxmtu 0x5DC
Mtu 1500, max_buffer_size 1524, max_pak_size 1608 enc 84
ROM rev: 0, FW OS rev: 0x00000000 Firmware rev: 0x00000000
 idb=0x42663A30, pa=0x427BF6E0, vip_fci_type=0, port_per_spa=0
 SPA port type is set
 Host SPI4 in sync
 SPA=0x427BF6E0 status=00010407, host=00000101, fpga=0x427EDF98
 cmd_head=113, cmd_tail=113, ev_head=184, ev_tail=184
 ev_dropped=0, cmd_dropped=0
```

```
!--- Start Link Record Information. tag 0, id 0, anyphy 0, anyphy_flags 3, state 0
crc 0, idle 0, subrate 0, invert 0, priority 0
encap hdlc
corrupt_ci 65535, transparent_ci 1
```

```
!--- End Link Record Information. Interface Serial6/0/0:0 is administratively down Channel
Stats: in_throttle=0, throttled=0, unthrottled=0, started=1 rx_packets=0, rx_bytes=0,
rx_frame_aborts=0, rx_crc_errors=0 rx_giants=0, rx_non_aligned_frames=0, rx_runts=0,
rx_overruns=0 tx_packets=0, tx_bytes=0, tx_frame_aborts=0 is_congested=0, mapped=1, is_isdn_d=0,
tx_limited=1 fast_if_number=15, fastsend=0x403339E4 map=0x7FFFFFFF, turbo_vector_name=Copperhead
to Draco switching lc_ip_turbo_fs=403A9EEC, lc_ip_mdcs=403A9EEC
```

Для CT3 необходимо получить `vc num`, который может быть получен из выходных данных `show interface` последовательный `CT3_interface_name`.

Теперь информация о CI может быть получена из консоли SPA. Сначала перенаправьте выходные данные команд консоли SPA к RP с командой `spa_redirect rp ct3_freedm336`.

`spa_ct3_test freedm` показывает, что `linkrec` команда `vc` показывает необходимую информацию о CI.

dMFR

Отладка Контрольного пути: **событие dmfr отладки**

Отладка пути данных: **пакеты dmfr отладки**

Путь данных и ошибочная отладка контрольного пути: **ошибка dmfr отладки**

Формирование дампа пакетов на основе CI: Посмотрите [dMLP](#).

dLFI

Отладка Контрольного пути: **событие dlfi отладки**

Отладка пути данных: **фрагменты dlfi отладки**

Путь данных и ошибочная отладка контрольного пути: **ошибка dlfi отладки**

dDDR

Нет никаких специальных команд отладки; необходимо использовать [отладки dMLP](#).

В случае dLFloLL, возможно, придется использовать и dMLP и отладки dLFI. Эти отладки не являются условным выражением и, следовательно, иницируют для всех связей (bundle).

Вопросы и ответы

1. **Что такое dMLP?**dMLP короток для Распределенного Протокола PPP (как сообщили в [RFC1990](#)). Эта функция поддерживается распределенными платформами, как Cisco серии 7500 и серии 7600. dMLP позволяет вам комбинировать линии T1/E1 — в VIP на маршрутизаторе Cisco серии 7500 или FlexWAN в маршрутизаторе серии "7600" — в связку (bundle), которая имеет объединенные линии T1/E1 пропускной способности нескольких интерфейсов T1/E1. Это позволяет клиентам увеличивать пропускную способность вне T1/E1 без потребности купить линию T3/E3.
2. **Что "распределено" в dMLP?**Термин "распределенный" подразумевает, что коммутация пакетов сделана VIP а не RSP. В чем причина? Возможности коммутации RSP скорее ограничены, и это имеет много более важных заданий, чтобы сделать. VIP,

являющийся способным к коммутируемым пакетам, разгружает это действие от RSP. На основе RSP Cisco IOS все еще управляет ссылками. Создание пучка и разрушение сделаны RSP. Кроме того, обработка уровня управления PPP все еще сделана RSP, включая обработку всех управляющих пакетов PPP (LCP, Аутентификация и NCP). Однако, как только связка (bundle) установлена, обработка пакетов MLP передана в VIP для коммутации встроенным ЦП. Механизм dMLP (на VIP) обрабатывает все процедуры MLP, включая фрагментацию, чередование, инкапсуляцию, распределение нагрузки среди сложных соединений, и сортировку и повторную сборку входящих фрагментов. Функции, сделанные VIP в 7500 системах, сделаны Flexwan/Enhanced-FlexWAN в 7600 основанных системах.

3. **Как я знаю, распределена ли связка (bundle) или нет?** Выполните команду **show ppp multilink** в консоли маршрутизатора: `Router# show ppp multilink`

```
Multilink1, bundle name is udho2
Bundle up for 00:22:46
Bundle is Distributed
174466 lost fragments, 95613607 reordered, 129 unassigned
37803885 discarded, 37803879 lost received, 208/255 load
0x4D987C received sequence, 0x9A7504 sent sequence
Member links: 28 active, 0 inactive (max not set, min not set)
  Se11/1/0/27:0, since 00:22:46, no frags rcvd
  Se11/1/0/25:0, since 00:22:46, no frags rcvd
!--- Output suppressed.
```

4. **Если я обновлю к RSP16 или Sup720, то моя производительность dMLP будет лучше?** Нет. Быстродействие коммутации dMLP (или любая распределенная функция) зависит от VIP или рассматриваемого FlexWAN. Например, производительность VIP6-80 будет лучше, чем производительность с VIP2-50.
5. **Который PAs я могу использовать с этой функцией?** PA-MC-T3 PA-MC-2T3 + PA-MC-E3 PA-MC-2E1 PA-MC-2T1 PA-MC-4T1 PA-MC-8T1 PA-MC-8E1 PA-MC-STM-1 PA-MC-8TE1 + PA-4T + PA-8T CT3 IP-50 (7500 только)
6. **Сколько ссылок может быть настроено в одиночной связке (bundle)?** Существует много фасетов к этому ответу. Основным узким местом является Питание ЦПУ линейной карты (VIP/FlexWAN/Enhanced-FlexWAN2). Жесткий предел является 56 ссылками на связку (bundle), но много раз вы не можете настроить тех многие (и иметь так много коммутации трафика), или из-за Питания ЦПУ или ограниченных буферов. Эти номера основываются на этой рекомендации (на основе ЦП и памяти на VIP/FlexWAN/Enhanced-FlexWAN2): VIP2-50 (w/SRAM 4 МБ) Max. T1s = 12 VIP2-50 (w/SRAM 8 МБ) Max. T1s = 16 VIP4-80 Max. T1s = 40 VIP6-80 Max. T1s = 40 FlexWAN Max. T1s = будет обновлен вскоре Max. E1 расширенного FlexWan = 21 E1 на отсек (объединяют 42 E1 на линейную карту),
7. **Существует ли изменение в производительности, если я настраиваю 3 связки (bundle) с 3 T1s каждый или 1 связкой (bundle) с 9 T1s?** Нет никакого изменения в производительности, как доказано на лабораторных испытаниях. Однако с большим числом T1s в одиночной связке (bundle) (говорят 24 или 28 T1s в одиночной связке (bundle)), существуют проблемы с исчерпыванием буферов. Ее наиболее рекомендуемое, что у вас не есть больше чем 8 участвующих соединений (T1/E1) в одиночной связке (bundle).
8. **Как пропускная способность связки (bundle) определена?** Пропускная способность связки (bundle) не должна быть настроена. Это - совокупная пропускная способность всех участвующих соединений. Если у вас есть 4 T1s в связке (bundle), то пропускная

способность связки (bundle) составляет 6.144 Мбит/с.

9. **Который лучше? Распределение нагрузки CEF или dMLP?** Нет никакого простого ответа к этому. Ваши потребности решают, какой лучше. **ПЛЮСЫ MLP:** Распределение нагрузки CEF применимо только к IP - трафику. MLP балансирует весь трафик, передаваемый по связке (bundle). MLP поддерживает заказ пакетов. Сам IP терпим к переупорядочению, таким образом, это может не иметь значения для вас; фактически, дополнительная стоимость, вовлеченная в поддержание упорядочения, может быть причиной избежать MLP. IP предназначен для сетей, которые могут отправить дейтаграммы не в порядке, и что-либо с помощью IP, как предполагается, в состоянии иметь дело с переупорядочением. Однако несмотря на этот факт, действительность - то, что переупорядочение может все еще изложить реальную проблему. MLP предоставляет единственное логическое соединение равноправной системе. QoS поддерживается на Многоканальных соединениях. MLP предоставляет динамические емкости полосы пропускания, как пользователь может добавить или удалить участвующие соединения на основе текущих потребностей. MLP может связать большее число ссылок, тогда как распределение нагрузки CEF ограничено 6 параллельными путями IP. Распределение нагрузки CEF на поток ограничивает максимальную пропускную способность любого данного потока к одному T1. Например, у клиентов, использующих голосовые шлюзы, может быть много вызовов с тем же источником и назначением и, следовательно, использовать только один путь. **CONS MLP:** MLP добавляет дополнительные издержки к каждому пакету или кадру MLP является сом интенсивной загрузкой ЦПУ; dMLP является интенсивным ЦП линейной карты.
10. **Как я могу настроить множественные связки (bundle) между двумя маршрутизаторами?** Многоканальный определяет, к какой связке (bundle) ссылка присоединится на основе названия и дискриминатора оконечной точки узла. Для создания множественных отдельных связок (bundle) между двумя системами стандартный метод должен вынудить некоторые ссылки определить себя по-другому. Рекомендуемый метод является использованием команды **названия ppp chap hostname**.
11. **У меня могут быть участвующие соединения от другого PAs?** Нет. Если вы хотите выполнить dMLP, то он не поддерживается. Однако, если участвующие соединения добавлены от другого PAs, то контроль дан к RSP и не dMLP больше. MLP все еще функционирует, но не стало преимуществ dMLP.
12. **Я могу смешать участвующие соединения от обоих отсеков?** Нет. Если вы хотите выполнить dMLP, то он не поддерживается. Однако, если участвующие соединения добавлены от другого PAs, то контроль дан к RSP, и это больше не dMLP. MLP все еще функционирует, но не стало преимуществ dMLP.
13. **У меня могут быть участвующие соединения через другие VIP или FlexWANs?** Нет. Если вы хотите выполнить dMLP, то он не поддерживается. Однако, если участвующие соединения добавлены от другого PAs, то контроль дан к RSP и не dMLP больше. MLP все еще функционирует, но не стало преимуществ dMLP.
14. **У меня могут быть участвующие соединения через другие порты от одиночного PA?** (Например, одно участвующее соединение от каждого порта CT3 PA-MC-2T3 +.) Да. Пока это от того же PA, нет никаких проблем.
15. **Я могу связать T3 или порты E3?** Нет. Только DS0, n*DS0, T1 и скорости E1 позволен с dMLP для 7500/VIP, 7600/FlexWAN, и 7600/FlexWAN2. **Примечание:** Распределенный протокол MLPPP поддерживается только для участвующих соединений, настроенных

в T1/E1 или низкоскоростных скоростях T1/E1. Канализируемый STM-1/T3/T1 взаимодействует также dMLPPP поддержки в T1/E1 или низкоскоростных скоростях T1/E1. Распределенный протокол MLPPP не поддерживается для участвующих соединений, настроенных в T3/E3 clear-channel или более высоких интерфейсных скоростях.

16. **Что такое “переупорядоченные” фрагменты?**Если полученный фрагмент или пакет не совпадают с ожидаемым порядковым номером, то счетчик `reordered` инкрементно увеличен. Для переменных размеров пакета это связано произойти. Для пакетов фиксированного размера это может также произойти, потому что драйвер Пенсильвании обрабатывает пакеты, которые получили на одной ссылке и не идут в не основе кольцевого алгоритма Round robin (как сделан в dMLP при передаче пакетов). Переупорядоченный не означает потерю пакета.
17. **Что такое “потерянные” фрагменты?**Каждый раз, когда фрагмент или пакет получены не в порядке, и вы находите, что неисправные фрагменты или пакеты получены на всех ссылках, счетчик `lost fragments` инкрементно увеличен. Другой случай - когда неисправные фрагменты сохранены в списке, и это достигает предела (решенный на основе SRAM VIP и независимо от того, что назначено для связки (bundle)), потерянный счетчик фрагментов инкрементно увеличен, и следующий порядковый номер в списке взят для обработки.
18. **Как dMLP обнаруживает потерянные фрагменты?**Порядковые номера: Если вы ждете фрагмента с порядковым номером N для поступления, и все ссылки получают фрагмент с порядковым номером выше, чем N, вы знаете, что фрагмент N должен быть потерян, потому что нет никакого способа, которым это могло по закону поступить позади более высоких пронумерованных фрагментов в ту же ссылку. Таймаут: Если вы будете находиться, слишком долго ожидая фрагмента, то вы в конечном счете объявите его, как потерян и перейдете. Переполнение буфера сборки: Если вы ждете фрагмента N для поступления, и между тем другие фрагменты (с порядковыми номерами выше, чем N) поступают в некоторые ссылки, то необходимо припарковать те фрагменты в буфере сборки, пока не обнаруживается фрагмент N. Существует предел тому, сколько можно буферизовать. Если пополнение буфера, вы снова объявляете фрагмент N, как потерян и продолжаете обрабатывать с тем, что находится в буфере.
19. **Что “потеряно полученное?”**Существует две возможных причины для потерянных полученных фрагментов или пакетов:Если полученный фрагмент или пакет вне окна диапазона ожидаемой последовательности, пакет отброшен путем маркировки это, как потерян получено.Если полученный фрагмент или пакет в окне диапазона ожидаемой последовательности, но вы не в состоянии выделить заголовок пакета `geragent` этот пакет, то пакет отброшен и отмеченный, как потерян получено.
20. **Шифрование поддерживается с dMLP?**Нет.
21. **Мы поддерживаем сжатие заголовка PFC?**Нет, не в распределенном пути. Маршрутизатору дальнего конца не рекомендуют настроить сжатие заголовка PFC, потому что мы переключаемся на нераспределенный режим, если мы принимаем сжатые кадры заголовка или пакеты. Если вы хотите продолжить рабочий dMLP, сжатие заголовка PFC должно быть отключено в обоих концах.
22. **Программное сжатие поддерживается с dMLP?**Нет, потому что программное сжатие не будет работать в распределенном пути.
23. **Фрагментация поддерживается на передающей стороне?**Не с dMLP Ванили. Нет никаких проблем с получением фрагментов с dMLP Ванили, но на передающей

стороне, фрагментация не происходит. Когда **ppp multilink interleave** настроен на интерфейсе dMLP, фрагментация передающей стороны поддерживается.

24. **Мы можем пропинговать участвующие соединения Пучка MLP?** Нет, вы не можете настроить IP-адрес на участвующих соединениях.
25. **Есть ли зависимость от MTU ссылки и размера фрагмента MLP?** Нет. Максимальный размер передаваемого блока данных не имеет никакого отношения к размеру фрагмента MLP кроме очевидного ограничения, что фрагмент MLP, как любой другой кадр, не может превысить максимальные размеры передаваемого блока данных Последовательных соединений.
26. **Действительно ли возможно настроить два Пучка MLP между одиночной парой маршрутизаторов?** Да, это возможно. Однако это могло привести к распределению нагрузки, которому повреждают. Может быть полезно на испытательных стендах, моделировать больше чем два маршрутизатора, использующие всего два маршрутизатора, но это не имеет никакого очевидного реального значения. Все ссылки, которые переходят к общему узлу, *должны* быть помещены в ту же связку (bundle). По определению связка (bundle) является набором ссылок, которые переходят к конкретному одноранговому узлу. "Узел" определен значениями Имени пользователя и Дискриминатора оконечной точки, которые он предлагает во время LCP и фаз проверки подлинности. При попытке создать множественные связки (bundle) между двумя маршрутизаторами, то это означает, что вы пытаетесь сделать каждую подмену маршрутизатора, как являющуюся больше, чем одиночный узел к его дубликату. Они должны определить себя соответственно.
27. **Участвующие соединения могут иметь другие алгоритмы организации очереди?** Все механизмы организации очередей, отнесенные к связке (bundle), должны быть применены на уровне связки (bundle) а не на уровне участвующего соединения. Однако настройка алгоритма очереди не должна влиять, как пакеты коммутированы из связки (bundle).
28. **Когда dMLP включен на Cisco 7500, почему tx-queue-limit к 26 как по умолчанию для участвующих соединений для многоканального соединения?** Для любого Последовательного интерфейса T1/E1 пропускной способности tx-queue-limit - приблизительно 4 или 5. При связывании T1s/E1 вместе в многоканальном пропускная способность увеличилась бы для связки (bundle). Поскольку коммутация имела бы место на основе пропускной способности интерфейса MLP, необходимо увеличить tx-queue-limit участвующих соединений. Только одно из участвующих соединений, названных основным соединением, используется для коммутации, поэтому, ее tx-queue-limit должен быть увеличен. Кроме того, это значение является эмпирическим, выбранным после тестирования и затем настройки на это значение. В целом развертывания имеют не больше чем 4 - 6 ссылок T1/E1 в связке (bundle). Значение 26 может обслужить 6 - 8 ссылок T1/E1 отлично, и следовательно это значение было выбрано.
29. **Что такое дифференциальная задержка и ее значение в реализации dMLP?** dMLP поддерживает дифференциальную задержку 30 мс. Это означало бы, получен ли фрагмент за один раз T, и он не работает (ожидание порядкового номера 100, но мы получили 101). Если бы порядковый номер 100 не получен до мс T+30, 100 был бы объявлен потерянным и если бы можно начать обрабатывать от 101, вы сделали бы это. В случае, если вы не можете запустить с 101 (если бы это - средний фрагмент), вы искали бы фрагмент, который имеет начать фрагмент (например, 104), и запустите оттуда.

30. Когда пакеты фрагментированы на уровне IP с многоканальным на 7500, что происходит? Если пакеты фрагментированы на уровне IP, то они транспортируются без повторной сборки в промежуточных переходах, но повторно собраны в маршрутизаторе назначения.
31. Когда пакеты фрагментированы на уровне MLP на 7500, что происходит? Если пакеты фрагментированы на уровне MLP и если повторно собранные пакеты больше, чем MRRU, то пакеты отброшены на многоканальном. Фрагментация стороны передачи поддерживается на dMLP только с dLFI. Пакеты фрагментированы на уровне MLP, только если packet_size больше, чем frag_size и меньше, чем MRRU. Если пакеты, которые передаются больше, чем MRRU и если он не фрагментирован на уровне IP, то другой конец отбрасывает весь пакетный размер, они не фрагментированы на уровне MLP, потому что пакеты являются больше, чем MRRU.
32. Как вычислен MRRU? MRRU вычислен согласно этому предпочтению: Для новых входящих участвующих соединений о MRRU снова выполняют согласование на уровне LCP согласно MRRU, настроенному на участвующих соединениях. Значение настроено на интерфейсе канала с командой `ppp multilink mrru interface`. Если не настроенный, значение наследовалось конфигурации команды `ppp multilink mrru` на родительском интерфейсе. Если оба значения присутствуют, значение интерфейса канала имеет приоритеты. MRRU по умолчанию 1524.

Усовершенствования отладки

Эти усовершенствования будут приведены в рабочее состояние в будущем. Планирование еще не должно завершать.

- Включите команду `debug frame-relay multilink` на LC.
- Улучшите текущие CLI отладки на номер интерфейса и заданный номер пакетов.
- Для dDDR еще не поддерживается функциональность QoS. Это может быть приведено в рабочее состояние только с надлежащей деловой ситуацией.

Дополнительные сведения

- [Dialer CEF](#)
- [Настройка конфигурации однорангового DDR с профилями программы для набора номера](#)
- [MPLS — поддержка протокола PPP](#)
- [Распределенный Многоканальный протокол "точка-точка" для маршрутизаторов Cisco серии 7500](#)
- [Распределенный многоканальный Frame Relay \(FRF.16\)](#)
- [Фрагментация распределенного канала и чередующийся по выделенным линиям](#)
- [Каналы VoIP over PPP с поддержкой средств QoS \(LLQ / IP RTP Priority, LFI, cRTP\)](#)
- [Технические примечания по поиску и устранению проблем - маршрутизатор Cisco серии 7500](#)
- [Страница технической поддержки маршрутизатора - Cisco Systems](#)
- [Cisco Systems – техническая поддержка и документация](#)